

02/09/2025

## Compte rendu TP Code.org

### 1. Analyse du problème

Le but du TP était de déplacer un personnage dans un labyrinthe en utilisant des instructions de programmation.

Au début, les parcours étaient simples, mais ils devenaient progressivement plus longs et complexes. Il fallait réfléchir à la **meilleure suite d'instructions** pour atteindre la fin sans erreur. Image ci-dessous : premiers déplacements simples.



Ces premiers niveaux m'ont permis de comprendre le fonctionnement des instructions de base comme “**avancer**” ou “**tourner**”.

## 2. Conception de la solution

Pour résoudre les niveaux, j'ai commencé par **planifier le chemin** avant de coder.

- Dans les niveaux 6 à 10, j'ai découvert le bloc **“répéter”** qui permet d'exécuter plusieurs fois la même instruction sans la réécrire. Image ci-dessous : utilisation du bloc “répéter”.(c'est le bloc rose)

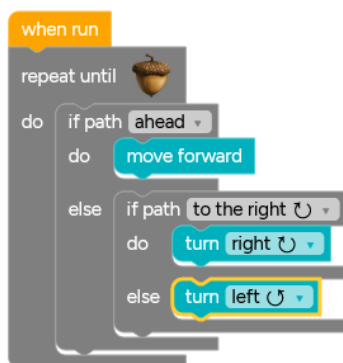


## 3. Réalisation

J'ai ensuite codé les solutions en suivant mes plans :

- Pour les premiers niveaux, j'ai utilisé des instructions simples.
- Pour les niveaux intermédiaires, j'ai utilisé les boucles pour simplifier et optimiser mon code.
- Pour les derniers niveaux, j'ai combiné toutes les méthodes pour trouver la **solution la plus efficace**.

*Image ci-dessous : exemple de labyrinthe complexe résolu.*



J'ai parfois eu des difficultés avec les directions gauche/droite ou quand je mettais trop d'instructions au lieu d'utiliser des boucles, mais j'ai ajusté mon code à chaque test.

## 4. Test

À chaque niveau, j'ai testé mon code pour vérifier que le personnage atteignait la sortie sans erreur.

Quand le personnage se trompait de direction, je revenais sur mon code, je corrigeais les instructions et je testais à nouveau.

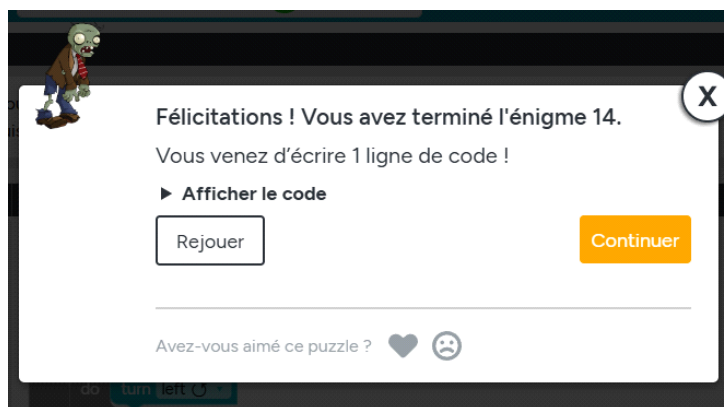
Cette phase m'a appris que **tester et corriger** est indispensable pour réussir en programmation.

## 5. Retour d'expérience

Ce TP m'a beaucoup appris :

- Être **plus logique** dans ma manière de réfléchir.
- Utiliser les **boucles** pour simplifier mon code et gagner du temps.
- **Planifier mes instructions** avant de les exécuter pour éviter les erreurs.

En conclusion, j'ai compris qu'en programmation, il est essentiel de **réfléchir avant de coder** et d'**optimiser ses actions** pour résoudre les problèmes efficacement.



A la fin de chaque niveau un message de félicitation s'affiche

